

manufacture including instruction means which implement the function/act specified in the block or blocks.

[0027] The computer program instructions may also be loaded onto a computer or other programmable data processing apparatus to cause a series of operational steps to be performed on the computer or other programmable apparatus to produce a computer implemented process such that the instructions which execute on the computer or other programmable apparatus provide steps for implementing the functions/acts specified in the block or blocks.

[0028] FIG. 1 illustrates a block diagram of a sample data-processing apparatus 100, which can be utilized for an improved contactless fingerprint scanner apparatus. Data-processing apparatus 100 represents one of many possible data-processing and/or computing devices, which can be utilized in accordance with the disclosed embodiments. It can be appreciated that data-processing apparatus 100 and its components are presented for generally illustrative purposes only and do not constitute limiting features of the disclosed embodiments.

[0029] As depicted in FIG. 1, a memory 105, a mass storage 107 (e.g., hard disk), a processor (CPU) 110, a Read-Only Memory (ROM) 115, and a Random-Access Memory (RAM) 120 are generally connected to a system bus 125 of data-processing apparatus 100. Memory 105 can be implemented as a ROM, RAM, a combination thereof, or simply a general memory unit. Module 111 includes software module in the form of routines and/or subroutines for carrying out features of the present invention and can be additionally stored within memory 105 and then retrieved and processed via processor 110 to perform a particular task. A user input device 140, such as a keyboard, mouse, or another pointing device, can be connected to PCI (Peripheral Component Interconnect) bus 145.

[0030] Data-processing apparatus 100 can thus include CPU 110, ROM 115, and RAM 120, which are also coupled to a PCI (Peripheral Component Interconnect) local bus 145 of data-processing apparatus 100 through PCI Host Bridge 135. The PCI Host Bridge 135 can provide a low latency path through which processor 110 may directly access PCI devices mapped anywhere within bus memory and/or input/output (I/O) address spaces. PCI Host Bridge 135 can also provide a high bandwidth path for allowing PCI devices to directly access RAM 120.

[0031] A communications adapter 155, a small computer system interface (SCSI) 150, and an expansion bus-bridge 170 can be attached to PCI local bus 145. The communications adapter 155 can be utilized for connecting data-processing apparatus 100 to a network 165. SCSI 150 can be utilized to control high-speed SCSI disk drive 160. An expansion bus-bridge 170, such as a PCI-to-ISA bus bridge, may be utilized for coupling ISA bus 175 to PCI local bus 145. Note that PCI local bus 145 can further be connected to a monitor 130, which functions as a display (e.g., a video monitor) for displaying data and information for a user and also for interactively displaying a graphical user interface (GUI) 185. A user actuates the appropriate keys on the GUI 185 to select data file options. Note that the term “GUI” generally refers to a type of environment that represents programs, files, options, and so forth by means of graphically displayed icons, menus, and dialog boxes on a computer monitor screen. A rendering device 190 can also be connected to the PCI local bus 145.

[0032] The embodiments described herein can be implemented in the context of a host operating system and one or

more modules. Such modules may constitute of hardware modules such as, for example, electronic components of a computer system. Such modules may also constitute software modules. In the computer programming arts, a software “module” can be typically implemented as a collection of routines and data structures that performs particular tasks or implements a particular abstract data type. Modules may be composed of two parts: an interface, which lists the constants, data types, variable, and routines that can be accessed by other modules or routines, and an implementation, which is typically private (accessible only to that module) and which includes source code that actually implements the routines in the module. The term module may also simply refer to an application such as a computer program designed to assist in the performance of a specific task such as word processing, accounting, inventory management, music program scheduling, etc.

[0033] FIG. 2 illustrates a schematic view of a software system 200 including an operating system, application software, and a user interface for carrying out the disclosed embodiments. Computer software system 200 directs the operation of the data-processing system 100 depicted in FIG. 1. Software application 202, stored in main memory 105 and on mass storage 107, includes a kernel or operating system 201 and a shell or interface 203. One or more application programs, such as software application 202, may be “loaded” (i.e., transferred from mass storage 107 into the main memory 105) for execution by the data-processing system 100. The data-processing system 100 receives user commands and data through the interface 203, as shown in FIG. 2. The user’s command input may then be acted upon by the data-processing system 100 in accordance with instructions from operating module 201 and/or application module 202.

[0034] The interface also serves to display and recognize, whereupon the user may supply additional inputs or terminate the session. In an embodiment, operating system 201 and interface 203 can be implemented in the context of a “Windows” system. It can be appreciated, of course, that other types of systems are possible. For example, rather than a traditional “Windows” system, other operation systems such as, for example, Linux may also be employed with respect to operating system 201 and interface 203. The software application 202 can include a contactless fingerprint scanning and recognition module 205. The software application 202 can also be configured to communicate with the interface 203 and various components and other modules and features as described herein.

[0035] FIG. 3 illustrates an exemplary contactless fingerprint scanner operating system 300, in accordance with the disclosed embodiments. System 300 includes: 1) a user interface 301 (e.g., on/off button, settings), 2) scanning device 302 (e.g., s/w & h/w—main component—has the “inner workings”), and 3) sensor 303 (e.g., collects the data from scanning. The sensor 303 can comprise a separate component, a remote sensor input, a base platform, dedicated space, or a light-source plus sensor, etc. The sensor 303 captures data from a scanned hand and orients the placement of the target to be scanned). The system 300 also includes: 4) output 304 (e.g., captured data, images, via—monitor display, printed, stored (disc, HD, ROM, etc.), processed, transferred, converted, measured, histogram, memory, etc.). The disclosed embodiments first validate a scanner’s ability to capture a simulated ridge pattern. A test pattern is created that presents a friction ridge pattern structure with a predictable geometric pattern to the scanner. Second, a scanner’s ability for capturing a simulated minutiae pattern is validated. A test pattern is